Freeform Search

Database: US Pre-Grant Publication Full-Text Database
US Patents Full-Text Database
US OCR Full-Text Database
EPO Abstracts Database
JPO Abstracts Database
Derwent World Patents Index
IBM Technical Disclosure Bulletins

Term: @pd<20030623 and L256

Display: 10 **Documents in Display Format:** - **Starting with Number** 1

Generate: ○ **Hit List** ◉ **Hit Count** ○ **Side by Side** ○ **Image**

Search Clear Interrupt

Search History

DATE: Monday, April 16, 2007 **Purge Queries** Printable Copy Create Case

Set Name side by side	Query	Hit Count	Set Name result set
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L257	@pd<20030623 and L256	21	L257
L256	l70 and L255	34	L256
L255	access$3 near ((shared or common) adj resource)	2938	L255
L254	L247 and L253	29	L254
L253	((global or local) near (pointer or variable)) with resource	172	L253
L252	@pd<20030623 and L251	23	L252
L251	L182 and L250	44	L251
L250	L247 and L249	191	L250
L249	(variable or pointer or address) near resource	3309	L249
L248	L246 and L247	15	L248
L247	access$3 near shared near (resource or memory)	5345	L247
L246	((thread or process or application) near (variable or pointer or address)) with resource	359	L246
L245	((exchan$3 or switch$3) near (variable or pointer or address)) with resource	54	L245
L244	((exchan$3 or switch$3) near (variable or pointer)) with resource	6	L244

L243	((shared near (memory or resource)) and (protect$3 or execlus$3)).ab,ti.	260	L243
	DB=TDBD,DWPI,JPAB,EPAB,USOC,USPT,PGPB; PLUR=YES; OP=OR		
L242	GOMES-CARLOS-P!	4	L242
L241	GOMES-CARLOS!	1	L241
L240	CHAN-HON-KEAT!	2	L240
L239	CHAN-HON-KEAT-W!	9	L239
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L238	L233 and L237	15	L238
L237	thread and (heap or (shared near (resource or memory or address)))	7943	L237
L236	thread and (heap or (shared near (resource or memory or address)))and L234	2155	L236
L235	L233 and L234	3	L235
L234	thread with (heap or (shared near (resource or memory or address)))	2155	L234
L233	((switch$3 or exchang$3) near variable)	8025	L233
L232	@pd<20030623 and L231	75	L232
L231	thread and L229	195	L231
L230	thead and L229	0	L230
L229	L224 and L228	354	L229
L228	wait$3 and L227	673	L228
L227	return$3 near pointer with (resource or address or memory or area or region)	1152	L227
L226	@pd<20030623 and L225	61	L226
L225	L223 and L224	97	L225
L224	(shared or common) near (resource or address or memory or area or region)	81273	L224
L223	return$3 near pointer with (local or global or shared)	180	L223
L222	L215 and L221	2	L222
L221	exchang$3 near pointer	191	L221
L220	exchang$3 with wait$3 near pointer	1	L220
L219	exchang$3 with local adj pointer	4	L219
L218	exchang$3 near local adj pointer	0	L218
L217	L215 and L216	34	L217
L216	pointer near (resource or memory or area or address or space or region)	21402	L216
L215	pointer near wait$3	153	L215
L214	(thread or task) and L213	63	L214
L213	pointer adj wait$3	94	L213
L212	pointer near wait$3 near (object or resource or component)	3	L212
L211	wait$3 near pointer	153	L211
L210	wait$3 near (pointer or variable)	557	L210
L209	local near (pointer or variable) near (thread or process)	145	L209
L208	thread near (swap$4 or switch$3 or exchang$3) near (pointer or address)	25	L208
L207	L205 and L206	23	L207
L206	(global or shared) near (pointer or variable) with ((shared or common or secure or protect$3) near (object or resource or memory or area or address or space or region))	392	L206

L205	local near (pointer or variable) with (thread or process)	817	L205
L204	L200 and L203	31	L204
L203	(exchang$3 or switch$3 or swap$4 or chang$3 or replac$6) near pointer	5657	L203
L202	L200 and L201	2	L202
L201	pointer near wait$3	153	L201
L200	pointer near ((shared or common or secure or protect$3) near (object or resource or memory or area or address or space or region))	277	L200
L199	pointer near ((shared or common or secure or protect$3) near (resource or memory or area or address or space or region))	244	L199
L198	5057996.pn.	2	L198
L197	(swap$4 or switch$3 or exchang$3) near (pointer or address) with ((shared or common or secure or protect$3) near (resource or memory or area or address or space or region))	105	L197
L196	indirect near pointer with (swap$4 or switch$3 or exchang$3)	4	L196
L195	indirect near pointer with (swap$4 or switch$3 or exchang$3) near (function or procedure or instruction)	0	L195
L194	(local with (global or shared) near pointer)	75	L194
L193	L45 and L192	19	L193
L192	(swap$4 or switch$3 or exchang$3) near (function or procedure or instruction) near address	185	L192
L191	chang$3 near local near pointer	8	L191
L190	((switch$3 or exchang$3) with ((local or global) near pointer))	31	L190
L189	L182 and L188	44	L189
L188	L162 and L176	311	L188
L187	L182 and L185	8	L187
L186	L182 aned L185	13628	L186
L185	jump$3 with ((local or global) near variable)	67	L185
L184	L172 and L162	11	L184
L183	L177 and L182	15	L183
L182	L178 or L179 or L180 or L181	10123	L182
L181	(719/312).ccls.	188	L181
L180	(709/213 |709/214 |709/215 |709/216).ccls.	2176	L180
L179	(711/147 |711/148 |711/149).ccls.	1928	L179
L178	(718/100 |718/101 |718/102 |718/103 |718/104 |718/105 |718/106 |718/107 |718/108).ccls.	6295	L178
L177	L172 and L176	142	L177
L176	(common or shared) near (resource or address or region or memory)	68224	L176
	DB=TDBD,DWPI,JPAB,EPAB,USOC,USPT,PGPB; PLUR=YES; OP=OR		
L175	GOMES-CARLOS!	1	L175
L174	CHAN-HON-KEAT-W!	9	L174
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L173	L171 and L172	11	L173
L172	((switch$3 or exchang$3) near pointer)	1212	L172

L171	pointer near wait$3	153	L171
	DB=USPT; PLUR=YES; OP=OR		
L170	'4543627'.pn.	1	L170
L169	'4744023'.pn.	1	L169
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L168	((switch$3 or exchang$3) near pointer) with (atomic$4 or interupt$3)	19	L168
L167	chang$3 near pointer with (local or global)	38	L167
L166	chang$3 near local near pointer with global	2	L166
	DB=USPT; PLUR=YES; OP=OR		
L165	'5950505'.pn.	1	L165
L164	'5892899'.pn.	1	L164
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L163	jump$3 near address and L162	54	L163
L162	(local and global) near (pointer or variable)	1139	L162
	DB=USPT; PLUR=YES; OP=OR		
L161	'5504930'.pn.	1	L161
L160	'5511207'.pn.	1	L160
L159	'5392207'.pn.	1	L159
L158	'5619409'.pn.	1	L158
L157	'5619409'.pn.	1	L157
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L156	((switch$3 or exchang$3) near (pointer or variable)) with ((shared or common or secure or protect$3) near (resource or memory or area or address or space))	28	L156
L155	L152 and L154	8	L155
L154	(exchang$3 or replac$6) near pointer	1404	L154
L153	L151 and L152	8	L153
L152	access near (shared or common) near resource	2780	L152
L151	pointer near wait$3	153	L151
L150	thread with access near resource and L149	9	L150
L149	pointer near (local$3 or global$3)	1216	L149
L148	L146 and L147	123	L148
L147	shared near (area or address or memory or resource)	39353	L147
L146	(replac$6 or switch$3 or exchang$3 or chang$3) with ((global or local) near (pointer or variable))	898	L146
L145	wait$3 near flag with (shared near (area or address or memory or resource))	3	L145
L144	interchangeabl$3 near pointer	19	L144
L143	L137 and L142	5	L143
L142	exclus$3 near access with (shared near (area or address or memory or resource))	347	L142
L141	return$ near (variable or pointer) with (shared near (area or address or memory or resource))	20	L141
L140	L136 and L139	1	L140

L139	pointer near wait$3	153	L139
L138	L136 and L137	24	L138
L137	((replac$6 or switch$3 or exchang$3 or chang$3) near pointer)	5476	L137
L136	pointer near (shared near (area or address or memory or resource))	178	L136
L135	initializ$5 near pointer with (shared near (area or address or memory or resource))	3	L135
L134	replac$6 near pointer with (shared near (area or address or memory or resource))	1	L134
L133	indirect near referenc$3 with (shared near (area or address or memory or resource))	1	L133
L132	indriect near referenc$3 with (shared near (area or address or memory or resource))	0	L132
L131	((switch$3 or exchang$3 or chang$3) near pointer) with (shared near (area or address or memory or resource))	8	L131
L130	indirect near pointer with (shared near (area or address or memory or resource))	4	L130
L129	L124 and L127	36	L129
L128	L112 and L127	104	L128
L127	initializ$4 near (global or local) near (pointer or variable)	298	L127
L126	L124 and L125	115	L126
L125	return near pointer with (area or address or memory or resource)	806	L125
L124	(exchang$3 or chang$3 or replac$6) near (pointer or variable)	29600	L124
L123	L120 and L122	1	L123
L122	branch$3 near (pointer or address) with access$3 near (area or address or memory or resource)	145	L122
L121	L119 and L120	103	L121
L120	(exchang$3 or chang$3 or replac$6) near pointer	4564	L120
L119	@pd<20030623 and L118	1637	L119
L118	pointer and L117	2260	L118
L117	L112 and L116	3753	L117
L116	branch$3 near (pointer or address)	6984	L116
L115	@pd<20030623 and L114	181	L115
L114	pointer and L113	327	L114
L113	L111 and L112	850	L113
L112	access$3 near (area or address or memory or resource)	409450	L112
L111	((thread or process) near (jump$3 or thunk$3))	3183	L111
L110	((jump$3 or thunk$3) with (local near variable))	27	L110
L109	((jump$3 or thunk$3) with (enter$4 near (area or address or memory or resource)))	90	L109
L108	L79 and L107	56	L108
L107	(swap$4 or switch$3 or exchang$3) with ((local or global or shared) near (variable or mutexes or semaphore))	254	L107

DB=PGPB; PLUR=YES; OP=OR

L106	US-20050060416-A1.did.	1	L106
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L105	(swap$4 or switch$3 or exchang$3) with ((local or global or shared) near (pointer))	54	L105
L104	(swap$4 or switch$3 or exchang$3) near ((local or global or shared) near (pointer))	3	L104
L103	((swap$4 or switch$3 or exchang$3) near (pointer)) with wait$3	7	L103
	DB=USPT; PLUR=YES; OP=OR		
L102	'5805900'.pn.	1	L102
L101	'6052695'.pn.	1	L101
L100	'6105085'.pn.	1	L100
L99	'6105085'.pn.	1	L99
	DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR		
L98	L74 and L97	59	L98
L97	((swap$4 or switch$3 or exchang$3) near (pointer))	1418	L97
L96	L79 and L95	35	L96
L95	((local and global) with (mutex or semaphore))	56	L95
L94	((local and global) near (mutex or semaphore))	13	L94
L93	L88 and L92	12	L93
L92	content$5 near (shared or common) near (resource or address or memory or area or region)	689	L92
L91	L74 and L90	2	L91
L90	L88 and L81	12	L90
L89	L87 and L88	4	L89
L88	pointer near (shared or common) near (resource or address or memory or area or region)	170	L88
L87	((switch$3 or exchang$3) near (function or procedure or method)) with (shared or common) near (resource or address or memory or area or region)	154	L87
L86	L74 and L85	67	L86
L85	L81 and L84	584	L85
L84	(thread or process) near wait$3	10253	L84
L83	(thread or process) near wait	7139	L83
L82	L74 and L81	1050	L82
L81	((switch$3 or exchang$3) near (function or procedure or method))	126303	L81
L80	L78 and L79	8	L80
L79	(shared or common) near (resource or address or memory or area or region)	81273	L79
L78	((switch$3 or exchang$3) near (function or procedure or method)) with (atomic$4 or without near interrupt$3)	134	L78
L77	L74 and L76	3	L77
L76	((switch$3 or exchang$3 or chang$3) near (function or procedure or method)) with (atomic$4 or without near interrupt$3)	243	L76
L75	L73 and L74	98	L75
	access$3 near (resource or ((shared or common) near (address or memory or		

L74	area or region)))	21513	L74
L73	((return$3 or relinquish$3) near pointer) with (resource or memory)	639	L73
L72	(pointer near (resource or wait$3))	668	L72
L71	L53 and L70	73	L71
L70	(local and (shared or global)) near (pointer or variable)	1286	L70
L69	L55 and L67	0	L69
L68	L53 and L67	30	L68
L67	(alter$3 or chang$3) near (local and (shared or global))	292	L67
L66	(alter$3 or chang$3) with (local and (shared or global))	3557	L66
L65	(alter$3 or chang$3) with (local and (shared or global)) near pointer	2	L65
L64	(swap$4 or switch$3 or exchang$3) with (local and (shared or global)) near pointer	3	L64
L63	718/???.ccls. and L61	6	L63
L62	L55 and L61	5	L62
L61	@pd<20030623 and L60	205	L61
L60	L53 and L59	306	L60
L59	(swap$4 or switch$3 or exchang$3) near (local or shared or global)	18403	L59
L58	pointer and L50	25	L58
L57	pointer and L55	1418	L57
L56	L50 and L55	1	L56
L55	(swap$4 or switch$3 or exchang$3) near pointer	1418	L55
L54	L50 and L53	6	L54
L53	(access$3) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	9240	L53
L52	L50 and L51	1	L52
L51	(pointer) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	235	L51
L50	(branch$3 or jump$3) near ((shared or common or protected) near (resource or address or region or portion or memory or area))	161	L50
L49	(branch$3 or jump$3) near ((shared or common or protected) near (address or region or portion or memory or area))	161	L49
L48	L45 and L47	21	L48
L47	(branch$3 or jump$3) near (swap$4 or switch$3 or exchang$3) near (function or procedure or instruction)	253	L47
L46	L44 and L45	46	L46
L45	((shared or common or protected) near (address or region or portion or memory or area))	114146	L45
L44	(swap$4 or switch$3 or exchang$3) near (function or procedure or instruction) with ((memory or address) near (location or pointer))	153	L44
L43	(swap$4 or switch$3 or exchang$3) near (function or procedure or instruction) near pointer	32	L43
L42	L37 and L41	90	L42
L41	(pointer or address) near wait$3	1126	L41

L40	(pointer or address) near wait	687	L40
L39	L37 and L38	52	L39
L38	(exchang$3 near (variable or pointer or address))	3016	L38
L37	(pointer or address) near ((shared or common) near (resources or address or region or memory or space) or heap)	7003	L37
L36	thread and L35	17	L36
L35	L33 and L34	133	L35
L34	(shared or common) near (address or region or memory or space) or heap	75352	L34
L33	(exchang$3 or switch$3) near (local or global or shared) with (variable or pointer or address)	914	L33
L32	(exchang$3 or switch$3) near (variable or pointer or address) with heap	4	L32
L31	L30 and L28	42	L31
L30	(exchang$3 or switch$3) near (tasks or thread)	6072	L30
L29	switch$3 near (tasks or thread)	5312	L29
L28	global adj variables and local adj variables	916	L28
L27	thread and L26	54	L27
L26	@pd<20030623 and L25	962	L26
L25	L21 and L24	1522	L25
L24	(exchang$3 or switch$3) near (pointer or address or variable)	20641	L24
L23	L21 and L22	1420	L23
L22	(exchang$3 or switch$3) near (pointer or address)	12703	L22
L21	(shared or common) near (address or region or memory or space)	61156	L21
L20	L14 and L19	47	L20
L19	thread with (heap or (shared near (resource or memory)))	2135	L19
L18	L14 and L17	1	L18
L17	thread near (heap or (shared near (resource or memory)))	230	L17
L16	thread and L15	117	L16
L15	L14 and (heap or (shared near (resource or memory)))	188	L15
L14	class near pointer	847	L14
L13	class with (property near pointer) and (heap or (shared near (resource or memory)))	4	L13
L12	((exchang$3 or switch$3) near (pointer or address)) with (heap or (shared near (resource or memory)))	16	L12
L11	L9 and L10	38	L11
L10	pointer near class	847	L10
L9	pointer near (heap or (shared near (resource or memory)))	403	L9
L8	thread near pointer with (heap or (shared near (resource or memory)))	2	L8
L7	thread near pointer near (function or method)	8	L7
L6	L4 and L5	27	L6
L5	request with (heap or (shared near (resource or memory)))	4359	L5
L4	thread with (local near (address or pointer or reference or variable))	400	L4
L3	(local and global) near (shared near resource)	3	L3

L2	((local or global) near (pointer or variable or address)) with (shared near resource)	25	L2
L1	20030061394	3	L1

END OF SEARCH HISTORY

CiteSeer Find: exchange variable shared resource Documents Citations

Searching for **PHRASE exchange variable shared resource.**
Restrict to: Header Title Order by: Expected citations Hubs Usage Date Try: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP
No documents match Boolean query. Trying non-Boolean relevance query.
500 documents found. **Only retrieving 250 documents (System busy - maximum reduced). Order: relevance to query.**

Resource Sharing in Reservation-Based Systems - de Niz,, Saewong, Rajkumar.. (2001) (Correct) (4 citations)
Resource Sharing in Reservation-Based Systems Dionisio de Niz,
Resource Sharing in Reservation-Based Systems Dionisio de
real-time operating systems began to support the **resource** reservation paradigm. This technique has proved
www.cs.cmu.edu/afs/cs/project/rtml-2/Papers/RTAS01/deniz.ps.gz

Modification of Shared Resources in AL-1/D - Hideaki Okamuray (Correct)
sweep, reference counting and so on. We want to **exchange** these algorithms and to evaluate new GC
line 5 to line 8. This meta object has internal **variables** such as gc, scheduler, allocator, segment,
Modification of **Shared Resources** in AL-1/D Hideaki Okamuray
jerry.cs.uiuc.edu/reflection/washington/postscript/okamura-letter.ps

A Comparative Study of Two Generic Resource Allocation Models - Youssef Abdel-Wahab (1997) (Correct)
arise in distributed systems is the allocation of **shared resources** among a group of clients. This problem
A Comparative Study of Two Generic **Resource** Allocation Models A. Youssef, H. Abdel-Wahab,
Norfolk, VA 23529 Abstract In a typical **resource** allocation model, clients request a fixed amount
www.cs.odu.edu/~youssef/papers/NC97.ps

A New Distributed Resource-Allocation Algorithm with.. - Sivilotti, Pike, Sridhar (2000) (Correct) (2 citations)
problem [6]Although rst postulated as a **shared**-memory concurrency problem, it has since received
A New Distributed **Resource**-Allocation Algorithm with Optimal Failure
problem {a classic problem in distributed **resource** allocation {that has optimal failure locality.
www.cis.ohio-state.edu/~nsridhar/Research/Papers/PS/fl.ps

Mutual Exclusion on Optical Buses - Kavi, Mehta (Correct)
contain the same values in each processor. The **variable** my turn[r] for a **resource** r in processor P i
survey)To the best of our knowledge, the **shared**-memory paradigm for parallel computing has not
it arises whenever concurrent access to **shared resources** by several sites is involved [4]For
crash1.eb.uah.edu/~kavi/Research/pdcs-00-mutex.pdf

Search Tree Unification: Paradigm for Process-based Logic.. - Ferenczi (Correct)
with the same communication pattern and unifying **variables** used in the individual communications [FF92] as
Recent blackboard oriented languages like **Shared** Prolog [BC91] belong to process-based languages
the program. This partition serves as **shared resource** for the processes, just like blackboards in
ftp.elis.rug.ac.be/pub/prolog/iclp94_workshop/ferenczi.ps.Z

The Mutual Exclusion Problem Has Been Solved - Lamport (1990) (Correct)
underlying mutual exclusion algorithms in **shared** memory systems is that: b. A memory reference to
executing processes do not access a **shared resource** at the same time. In a **shared** memory system,
memory system, each word of memory is a **shared resource**. Assumption b states that two concurrently
research.compaq.com/SRC/personal/lamport/pubs/lamport-mutual-solved.ps.Z

A Survey of Distributed Mutual Exclusion Algorithms - Velazquez (1993) (Correct) (3 citations)
algorithms, failures in the system need the **exchange** of more information than in normal operation, in
finite amount of time, but the time of arrival is **variable**. The topology of the network is known. Nodes
distributed computing systems are many. **Resource shar**ing, parallel processing, system availability and
www.cs.colostate.edu/~ftppub/TechReports/1993/tr-116.ps.Z

Local Serialization Pattern - Silva, Pereira, Sousa (Correct)
other concurrent access modes, e.g. if a **shared variable** is being written it cannot be simultaneously

for the concurrency control concern within a **shared resource**. 1 Introduction In this paper we
the concurrency control concern within a **shared resource**. 1 Introduction In this paper we present the
siesta.cs.wustl.edu/~schmidt/OOPSLA-95/html/papers/atomobj.ps.gz

Verifying Mutual Exclusion and Liveness Properties with TLA - Pau, Rents, Schreiner (2000) (Correct)
In this algorithm we use the following **shared variables**: turn 2 f0 1g f lag i 2 f0 1g where i 2
the following problem: given two processes and a **shared resource**, which must not be accessed by these two
problem: given two processes and a **shared resource**, which must not be accessed by these two
www.risc.uni-linz.ac.at/people/schreine/papers/extla.ps.gz

Yet Another Distributed Mutual Exclusion Algorithm - Teixeira, de Andrade.. (Correct)
of a single token. The average number of messages **exchanged** per critical section is O(log N)where N is
its parent. For instance, the node i maintains a **variable** parent i which indicates its parent node.
two or more nodes which simultaneously request the **shared resource**. Such queue is controlled by the current
www.cs.umd.edu/~hcma/papers/sccc95.ps.gz

Tight Space Self-stabilizing Uniform l-Mutual Exclusion - Gradinariu, Tixeuil (2000) (Correct)
where the processors maintain two types of **variables**: local **variables** and eld **variables**. The local
exclusion :the system has to guarantee the fair **shar**ing of a **resource** that can be used by l processors
the system has to guarantee the fair **shar**ing of a **resource** that can be used by l processors
www.lri.fr/~mariag/icdcs01.ps

Fair On-Line Scheduling of a Dynamic Set of Tasks on a.. - Baruah, Gehrke, Plaxton (1996) (Correct)
(12 citations)
)Otherwise, we say that the task requests are **variable**. The total request at slot t, denoted R(t)is
processors, disks, network bandwidth) may be **shared** among different "tasks" e.g.user processes)
Scheduling of a Dynamic Set of Tasks on a Single **Resource** Sanjoy K. Baruah Johannes E. Gehrke y C.
www.cs.wisc.edu/~johannes/papers/tr96-03.ps

Non-Preemptive Real-Time Scheduling Of Dataflow Systems - Parks, Lee (1995) (Correct)
enabled subtasks. An activation frame holds local **variables** that are **shared** by the subtasks. When a
include tasks that contend for exclusive access to **shared resources**. We show that non-preemptive
tasks that contend for exclusive access to **shared resources**. We show that non-preemptive rate-monotonic
ptolemy.eecs.berkeley.edu/~parks/papers/icassp95-parks/icassp95.ps.Z

Bandwidth Regulation of Real-Time Traffic Classes in.. - Liebeherr, Akyildiz.. (1994) (Correct) (2 citations)
Rate p (gw:li) and OL p (gw:li)and two **variables**, **Share** p (gw:li) and Surplus p (gw:li) for
has identical bandwidth constraints if their routes **share** the link with the smallest capacity for this
solution is appropriate if sufficient network **resources** are always available. Additionally, one may
ftp.cs.virginia.edu/pub/jorg/papers/CS-94-24.ps

Some Performance Issues Associated with CEWES MSRC Scalable.. - Bova (1997) (Correct)
product requires point-topoint communication for **exchange** of data on the processor interfaces, and the dot
Computing Modernization Program CEWES Major **Shared Resource** Center through Programming Environment
Modernization Program CEWES Major **Shared Resource** Center through Programming Environment and
www.wes.hpc.mil/pet/tech_reports/reports/pdf/tr_9702.pdf

The ccNUMA Memory Profiler - Alan Davis Ald (Correct)
increasingly popular. Compared to the traditional **shared**-bus (SMP) systems, they scale to much higher
scalability is not limited by a single **shared resource**. Compared to other scalable multiprocessor
ccNUMA designs replace one **shared resource** (system bus) with a collection of distributed
www.cs.utah.edu/~ald/pubs/CC-numa.pdf

First 20 documents Next 20

Try your query at: Google (CiteSeer) Google (Web) Yahoo! MSN CSB DBLP

Displaying records #491 through 500 out of 500
(search stopped at 500 hits)

Result # 491 Relevance:

Lock-less Concurrent Dual Chaining of Control Structures

2005-07-18 IPCOM000126443D En

The design required being able to chain a single element on two independent linked-list however there was an additional requirement that NO lock be obtained to perform the l management. The solution involves creating new control structures ...

Result # 492 Relevance:

Common Language-Independent Interface for Culturally Sensitive Fu

1994-07-01 IPCOM000113046D En

The disclosed method allows common interfaces for culturally sensitive functions to be application internationalization. These can be used across operating platforms and prog languages to support multiple cultures without the need to change the operation ...

Result # 493 Relevance:

Overlapped Field Processing

1972-06-01 IPCOM000077173D En

Certain programming languages allow data structures to be defined in such a manner tl characters or bits may be subdivided into overlapping subfields or substrings. During th compilation of a program written in such languages, it is first necessary ...

Result # 494 Relevance:

Floating Dialog Window for Presentation Manager

1991-10-01 IPCOM000121964D En

This article discusses a technique to display window information using the mouse pointe information can be obtained via mouse pointer position and displayed in an information information window is positioned opposite the mouse pointer, all visible ...

Result # 495 Relevance:

Simple Cryptographic Program Interface (Crypto API) (RFC2628)

1999-06-01 IPCOM000003215D En

This document describes a simple Application Program Interface to cryptographic functi purpose of such an interface is to separate cryptographic libraries from internet applica allowing an independent development of both. It can be used in various ...

Result # 496 Relevance:

Method of Configuring a Dynamic Mouse Pointer

1993-06-01 IPCOM000105211D En

Prior art allows certain information to be displayed dynamically in the window title bar a concatenation of the window title and pre-configured information. This prior art include: information such as incoming mail status or a search results list into the ...

Result # 497 Relevance:

Module to Module Linkage Mechanism

1982-12-01 IPCOM000050907D En

This article presents definition and construction of resource manager to resource mana functions. Reference to application to resource manager linkage services is made only v service function is common to both types of linkage.

Result # 498 Relevance:

Hypertext Capability for Interactive Edit of Program Source

1997-05-01 IPCOM000118699D En

Disclosed is a mechanism for introducing a hypertext capability for processing program during an interactive program edit.

Result # 499 Relevance:

Object selection method by dynamically changing units of selection

2002-10-10 IPCOM000015942D En

Disclosed is a solution for an object selection method for a software application. In a tr software application, the common way for object selection using a mouse requires deci starting and ending points and placing the mouse pointer at both points ...

Result # 500 Relevance:

A whole program register promotion method for global variables

2003-05-21 IPCOM000012694D En

This is a method to allocate registers to global variables through the entire program.

Displaying page 50 of 50 << FIRST | < BACK | NEXT > | LAST >>

Search query: exchange function comprising changing a local pointer for a global poi

New search | Modify this search

Google Web Images Video News Maps more »

shared resource "exchanging variable " Search Patents Advanced Patent Se Google Patent Searc

Patents

Patents **1 - 2** on **shared resource "exchanging variable "**. (**0.13** seconds)

Method and apparatus for **exchanging variable** length frames by fixed length cell handling exchange
US Pat. 5610918 - Filed Jan 31, 1994 - Fujitsu Limited
Also where the **resource** of the ATM exchange 200 is **shared** commonly by the ATM terminals and the frame relay terminals, control of the ATM terminal ...

Multiprocessor system having statically determining **resource** allocation schedule at compile time and the using of static schedule with processor signals to control the execution time dynamically
US Pat. 5367678 - Filed Dec 6, 1990 - The Regents of the University of California
The STATICALLY DETERMINING **RESOURCE** other processor reads the ... particular data set is ready for **shared** memory facilitates **exchanging variable** amounts the ...

shared resource "exchanging variable " Search Patents

Google Patent Search Help | Advanced Patent Search

Google Home - About Google - About Google Patent Search

Patents

Your search - **shared resource "exchanging pointer"** - did not match any documents.

Suggestions:

- Make sure all words are spelled correctly.
- Try different keywords.
- Try more general keywords.
- Try fewer keywords.
- Try your search at www.uspto.gov